Flow of Earthquake App

Web User

FrontEnd

Server

Elastic_Search

Web User → FrontEnd:
```
Request
Passes Search parameters
1)Type
2)Magnitude
3)Location
4)Range
5)Sort_option
```

Server → Elastic_Search: Aunthenticating Identity

Elastic_Search → Server: Authenticated Id

FrontEnd → Server:
```
Request
axios.request(results)
```

```
const results = {
    method: 'GET',
    url: 'http://localhost:3001/results',
    params: {
      type: chosenType,
      mag: chosenMag,
      location: chosenLocation,
      dateRange: chosenDateRange,
      sortOption: chosenSortOption,
    },
```

Server → Elastic_Search:
```
Request
await client.search
( {
index: 'earthquakes',
body: {..}
} )
```

```
const body = await client.search({
    index: 'earthquakes',
    body: {
      sort: [
          { mag: {order: passedSortOption,},},
        ],
      size: 300,
      query: {
        bool: {
          filter: [
          {term: { type: passedType },},
          {range: { mag: {gte: passedMag,},},},
          {match: { place: passedLocation },},
          {range: {
              gte: `now-${passedDateRange}d/d`,
              lt: 'now/d',
            },},},],},},});
```

Elastic_Search → Server:
```
Response
Fetches relevant documents
based on the search query
```

Server → FrontEnd:
```
Response
res.json(body.hits.hits);
```

FrontEnd → Web User:
```
Response
Displays fetched documents
```
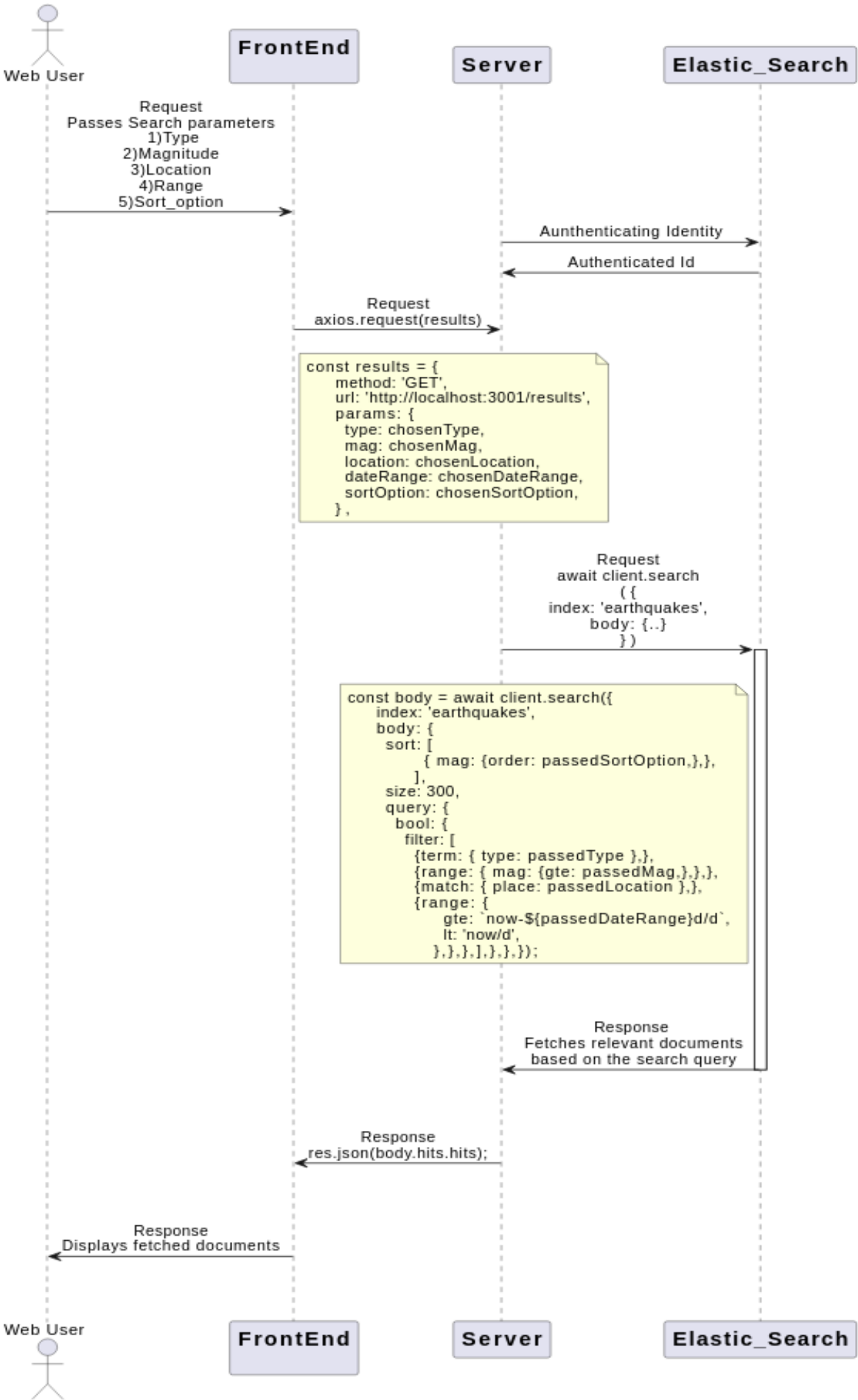
Web User

FrontEnd

Server

Elastic_Search

Flow of Earthquake App

Ingesting of data and indexing of documents into elastic search from earthquake API